

FILE ID DIR SCN

J 4

DISF
V04-

i RC

123456789101112131415161718191A1B1C1D1E1F1G1H1I1J1K1L1M1N1O1P1Q1R1S1T1U1V1W1X1Y1Z1
0001 0 MODULE DIRSCN (LANGUAGE (BLISS32) .
0002 0 IDENT = 'V04-000' .
0003 0) =
0004 1 BEGIN
0005 1
0006 1 *****
0007 1 *
0008 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0009 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0010 1 * ALL RIGHTS RESERVED.
0011 1 *
0012 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0013 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0014 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0015 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0016 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0017 1 * TRANSFERRED.
0018 1 *
0019 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0020 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0021 1 * CORPORATION.
0022 1 *
0023 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0024 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0025 1 *
0026 1 *
0027 1 *****
0028 1 ++
0029 1
0030 1
0031 1 FACILITY: F11ACP Structure Level 1
0032 1
0033 1 ABSTRACT:
0034 1
0035 1 This routine performs the basic scan of a directory,
0036 1 searching for both relevant entries and free space.
0037 1 ENVIRONMENT:
0038 1
0039 1 STARLET operating system, including privileged system services
0040 1 and internal exec routines.
0041 1
0042 1 --
0043 1
0044 1
0045 1
0046 1 AUTHOR: Andrew C. Goldstein, CREATION DATE: 26-Dec-1976 19:13
0047 1
0048 1 MODIFIED BY:
0049 1
0050 1 V03-004 LMP0281 L. Mark Pilant, 23-Jul-1984 8:43
0051 1 Fix a bug introduced by LMP0255 that caused null directory
0052 1 records to generate the SSS_BADIRECTORY error.
0053 1
0054 1 V03-004 LMP0255 L. Mark Pilant, 6-Jun-1984 10:32
0055 1 Validate the version number from a directory record to
0056 1 insure that it is greater than zero.
0057 1

58 0058 1 | V03-003 LMP0241 L. Mark Pilant, 26-Apr-1984 16:27
59 0059 1 | Change the wildcard handling logic to improve performance.
60 0060 1 |
61 0061 1 | V03-002 LMP0196 L. Mark Pilant, 28-Feb-1984 11:06
62 0062 1 | Only use FMGSMATCH_NAME for the file name and type. Not the
63 0063 1 | version.
64 0064 1 |
65 0065 1 | V03-001 LMP0167 L. Mark Pilant, 10-Nov-1983 14:23
66 0066 1 | Add support for full wildcarding.
67 0067 1 |
68 0068 1 | A0100 ACG0001 Andrew C. Goldstein, 10-Oct-1978 20:01
69 0069 1 | Previous revision history moved to F11A.REV
70 0070 1 |
71 0071 1 | **
72 0072 1 |
73 0073 1 | LIBRARY 'SYSSLIBRARY:LIB.L32';
74 0074 1 |
75 0075 1 | REQUIRE 'SRC\$:FCPDEF.B32';
76 0390 1 |

```
78 0391 1 GLOBAL ROUTINE DIR_SCAN (NAME_, OCK, ENTER_MODE) =  
79 0392 1 !++  
80 0393 1  
81 0394 1 FUNCTIONAL DESCRIPTION:  
82 0395 1  
83 0396 1  
84 0397 1 This routine performs the basic scan of a directory,  
85 0398 1 searching for both relevant entries and free space.  
86 0399 1  
87 0400 1 CALLING SEQUENCE:  
88 0401 1 DIR_SCAN (ARG1, ARG2)  
89 0402 1  
90 0403 1 INPUT PARAMETERS:  
91 0404 1 ARG1: address of name block  
92 0405 1 ARG2: 0 if operation is FIND or REMOVE  
93 0406 1 1 if operation is ENTER  
94 0407 1  
95 0408 1 IMPLICIT INPUTS:  
96 0409 1 NONE  
97 0410 1  
98 0411 1 OUTPUT PARAMETERS:  
99 0412 1 NONE  
100 0413 1  
101 0414 1 IMPLICIT OUTPUTS:  
102 0415 1 FIRST_FREE: record number of first free directory entry  
103 0416 1 DIR_RECORD: record number of found entry  
104 0417 1 HIGHEST_VERSION: highest version number encountered  
105 0418 1 LOWEST_VERSION: lowest version number encountered  
106 0419 1  
107 0420 1 ROUTINE VALUE:  
108 0421 1 address of found directory entry or  
109 0422 1 0 if not found  
110 0423 1  
111 0424 1 SIDE EFFECTS:  
112 0425 1 NONE  
113 0426 1  
114 0427 1 --  
115 0428 1  
116 0429 2 BEGIN  
117 0430 2  
118 0431 2 MAP  
119 0432 2 NAME_BLOCK : REF BBLOCK; ! name block arg  
120 0433 2  
121 0434 2 LINKAGE L_MATCH_NAME = JSB (REGISTER = 2, REGISTER = 3,  
122 0435 2 REGISTER = 4, REGISTER = 5);  
123 0436 2 : NOTUSED (10, 11);  
124 0437 2  
125 0438 2  
126 0439 2 EXTERNAL  
127 0440 2 FIRST_FREE, ! record number of first free slot  
128 0441 2 HIGHEST_VERSION, ! highest version number seen  
129 0442 2 LOWEST_VERSION, ! lowest version number seen  
130 0443 2 DIR_RECORD; ! record number of found entry  
131 0444 2  
132 0445 2 EXTERNAL ROUTINE  
133 0446 2 DIRGET, ! get a directory record  
134 0447 2 MAKE_STRING, ! convert file name block to string
```

```
135 0448 2 FMGSMATCH_NAME : L_MATCH_NAME;      ! match general wild card string
136 0449 2
137 0450 2 LOCAL
138 0451 2 HIG_VERSION : SIGNED WORD;        ! highest version so far
139 0452 2 LOW_VERSION : SIGNED WORD;         ! lowest version so far
140 0453 2 FREE_RECORD,                      ! first free record encountered
141 0454 2 REC_NUM,                          ! current record number
142 0455 2 BEST_REC,                         ! record number of highest/lowest version
143 0456 2 RECAADDR : REF BBLOCK;           ! address of current directory record
144 0457 2 ASCII_NAME : VECTOR [20, BYTE];   ! Dir entry name in ASCII
145 0458 2
146 0459 2 ! Initialize things.
147 0460 2
148 0461 2
149 0462 2 FREE_RECORD = 0;                  ! no empty record found yet
150 0463 2 BEST_REC = 0;                    ! no match found at all
151 0464 2 HIGH_VERSION = 0;                ! no high version as yet
152 0465 2 LOW_VERSION = 32767;            ! no low version as yet
153 0466 2
154 0467 2 ! Now scan the directory sequentially, looking at each entry. If the search
155 0468 2 is for *.*.*., then read one block at a time. Otherwise, read whatever the
156 0469 2 buffer pool will take.
157 0470 2 !
158 0471 2
159 0472 2 REC_NUM = .NAME_BLOCK [NMBSW_CONTEXT];
160 0473 2 IF
161 0474 3 BEGIN
162 0475 3 WHILE 1 DO
163 0476 4 BEGIN
164 0477 4 REC_NUM = .REC_NUM + 1;
165 0478 4 IF .NAME_BLOCK [NMBSV_ALLNAM]
166 0479 4 AND .NAME_BLOCK [NMBSV_ALLTYP]
167 0480 4 AND .NAME_BLOCK [NMBSV_ALLVER]
168 0481 4 THEN RECAADDR = DIRGET (.REC_NUM, 0)
169 0482 4 ELSE RECAADDR = DIRGET (.REC_NUM, 1);
170 0483 4
171 0484 4 IF .RECAADDR EQ 0 THEN EXITLOOP 1;    ! out on end of file
172 0485 4
173 0486 4 ! Process each directory entry. First check for empty entries, noting
174 0487 4 the first one.
175 0488 4 !
176 0489 4
177 0490 4 IF .RECAADDR [NMBSW_FID_NUM] EQ 0
178 0491 4 THEN
179 0492 5 BEGIN
180 0493 5 IF .FREE_RECORD EQ 0 THEN FREE_RECORD = .REC_NUM
181 0494 5 END
182 0495 5
183 0496 5 ! If we are in file ID search mode, compare the file ID
184 0497 5 !
185 0498 5
186 0499 4 ELSE
187 0500 5 BEGIN
188 0501 5 IF .RECAADDR [NMBSW_VERSION] LEQ 0 THEN ERR_EXIT (SSS_BADIRECTORY);
189 0502 5 IF .NAME_BLOCK [NMBSV_FINDFID]
190 0503 5 THEN
191 0504 6 BEGIN
```

```
192 0505 6 IF CHSEQL (NMBSS_FID, NAME_BLOCK [NMB$W_FID],  
193 0506 6 NMBSS_FID, RECADDR [NMB$W_FID], 0)  
194 0507 6 THEN EXITLOOP 0;  
195 0508 6 END  
196 0509 6  
197 0510 6  
198 0511 6 ! Otherwise compare the name and type fields under wild card control.  
199 0512 6 !  
200 0513 6  
201 0514 5 ELSE IF  
202 0515 5 BEGIN  
203 0516 6 IF .NAME_BLOCK[NMB$V_ALLNAM] AND .NAME_BLOCK[NMB$V_ALLTYP]  
204 0517 6 THEN 1  
205 0518 6 ELSE IF .NAME_BLOCK[NMB$V_WILD]  
206 0519 6 THEN  
207 0520 6 BEGIN  
208 0521 7 ASCII_NAME[0] = MAKE_STRING (.RECADDR, ASCII_NAME[1]);  
209 0522 7  
210 0523 7  
211 0524 7 ! Note that the match takes place ONLY on the file name and type.  
212 0525 7  
213 0526 7 ASCII_NAME[0] = CH$IND_CH (.ASCII_NAME[0],  
214 0527 7 ASCII_NAME[1], ';') -  
215 0528 7 ASCII_NAME[1];  
216 0529 7 FMG$MATCH_NAME (.ASCII_NAME[0],  
217 0530 7 ASCII_NAME[1],  
218 0531 7 .NAME_BLOCK[NMB$B_ASCNAMSIZ],  
219 0532 7 NAME_BLOCK[NMB$T_ASCNAMTXT])  
220 0533 7  
221 0534 6 ELSE END  
222 0535 7  
223 0536 7 (CHSEQL (NMBSS_NAME, NAME_BLOCK [NMB$W_NAME],  
224 0537 7 NMBSS_NAME, RECADDR [NMB$W_NAME], 0);  
225 0538 6 AND .NAME_BLOCK [NMB$W_TYPE] EQL .RECADDR [NMB$W_TYPE])  
226 0539 6  
227 0540 6 ! On a name and type match, process the version. Wild card is an immediate  
228 0541 6 match. Otherwise, maintain highest and lowest numbers found, keeping  
229 0542 6 the record number of what we are looking for, if any. On exact match,  
230 0543 6 exit immediately, note, or ignore, depending on the mode.  
231 0544 6 !  
232 0545 6  
233 0546 5 THEN  
234 0547 6 BEGIN  
235 0548 6 IF .NAME_BLOCK [NMB$V_ALLVER] THEN EXITLOOP 0; ! wild card version  
236 0549 6  
237 0550 6 IF .RECADDR [NMB$W_VERSION] GTR .HIGH_VERSION  
238 0551 6 THEN  
239 0552 7 BEGIN  
240 0553 7 HIGH_VERSION = .RECADDR [NMB$W_VERSION];  
241 0554 7 IF NOT .ENTER_MODE AND .NAME_BLOCK [NMB$W_VERSION] EQL 0  
242 0555 7 THEN BEST_REC = .REC_NUM;  
243 0556 6 END;  
244 0557 6  
245 0558 6 IF .RECADDR [NMB$W_VERSION] LSS .LOW_VERSION  
246 0559 6 THEN  
247 0560 7 BEGIN  
248 0561 7 LOW_VERSION = .RECADDR [NMB$W_VERSION];
```

```

: 249      0562 7           IF NOT .ENTER_MODE AND .NAME_BLOCK [NMBSW_VERSION] EQL -32768
: 250      0563 7           THEN
: 251      0564 7           BEST_REC = .REC_NUM;
: 252      0565 6           END;
: 253      0566 6
: 254      0567 6           IF .RECADDR [NMBSW_VERSION] EQL .NAME_BLOCK [NMBSW_VERSION]
: 255      0568 6           THEN
: 256      0569 7           BEGIN
: 257      0570 7           IF NOT .ENTER_MODE THEN EXITLOOP 0;
: 258      0571 7           IF NOT .NAME_BLOCK [NMBSV_NEVER] THEN BEST_REC = .REC_NUM;
: 259      0572 6           END;
: 260      0573 6
: 261      0574 5           END:                                ! end of version processing
: 262      0575 4           END END:                            : R
: 263      0576 4           END:                                ! end of loop
: 264      0577 3           END:                                end surrounding loop
: 265      0578 2           THEN END:                            if we scanned the whole directory
: 266      0579 2           IF .BEST_REC NEQ 0                and found a candidate
: 267      0580 2           THEN
: 268      0581 3           BEGIN
: 269      0582 3           RECADDR = DIRGET (.BEST_REC, 0);
: 270      0583 3           REC_NUM = .BEST_REC;
: 271      0584 2           END;
: 272      0585 2
: 273      0586 2           ! Return the implicit outputs as specified and return the address of the
: 274      0587 2           ! found directory entry (or 0 for none).
: 275      0588 2
: 276      0589 2
: 277      0590 2           FIRST_FREE = .FREE_RECORD;          ! first free entry
: 278      0591 2           HIGHEST_VERSION = .HIGH_VERSION;    ! highest version number found
: 279      0592 2           LOWEST_VERSION = .LOW_VERSION;     ! lowest version number found
: 280      0593 2           DIR_RECORD = .REC_NUM;          ! record number of entry
: 281      0594 2           RETURN .RECADDR;                  ! end of routine DIR_SCAN
: 282      0595 1           END;

```

```
.TITLE DIRSCN
.IDENT \V04-000\
```

```
.EXTRN FIRST_FREE, HIGHEST_VERSION
.EXTRN LOWEST_VERSION, DIR_RECORD
.EXTRN DIRGET, MAKE_STRING
.EXTRN FMGSMATCH_NAME
```

```
.PSECT SCODES,NOWRT,2
```

```
.ENTRY DIR_SCAN, Save R2,R3,R4,R5,R6,R7,R8,R9,R10,-, 0391
R11
```

			OFFC 00000	
	5E		1C C2 00002	SUBL2 #28 SP
		04	5A 7C 00005	CLRQ BEST_REC
	6E	7FFF	AE B4 00007	CLRW HIGH_VERSION
	50	04	B0 0000A	MOVW #32767, LOW_VERSION
	59	26	AC D0 0000F	MOVL NAME_BLOCK, R0
		59	A0 3C 00013	MOVZWL 3B(R0), REC_NUM
		04	D6 00017 1\$:	INCL REC_NUM
OE	10	50	AC D0 00019	MOVL NAME_BLOCK, R0
		04	05 E1 0001D	BBC #5, T6(R0), 2\$

09	10	A0	04	E1 00022	BBC	#4, 16(R0), 2\$: 0479
04	10	A0	03	E1 00027	BBC	#3, 16(R0), 2\$: 0480
			7E	D4 0002C	CLRL	-(SP)	: 0481
			02	11 0002E	BRB	3\$: M
			01	DD 00030	PUSHL	#1	: C
			59	DD 00032	PUSHL	REC_NUM	
		0000G CF	02	FB 00034	CALLS	#2, DIRGET	
		57	50	DO 00039	MOVL	R0, RECADDR	
			03	12 0003C	BNEQ	4\$	
			00CC	31 0003E	BRW	15\$	
			67	B5 00041	TSTW	(RECADDR)	
			09	12 00043	BNEQ	5\$	
			5B	D5 00045	TSTL	FREE_RECORD	
			CE	12 00047	BNEQ	1\$	
			59	D0 00049	MOVL	REC_NUM, FREE_RECORD	
			C9	11 0004C	BRB	1\$	
			0E	A7 0004E	TSTW	14(RECADDR)	
			05	14 00051	BGTR	6\$	
			0828	8F BF 00053	CHMU	#2088	
				04 00057	RET		
			56	04 AC 00058	MOVL	NAME_BLOCK, R6	
			58	10 A6 0005C	MOVAB	16(R6), R8	
			68	0B E1 00060	BBC	#11, (R8), 7\$	
			66	06 29 00064	CMPC3	#6, (R6), (RECADDR)	
			AD	12 00068	BNEQ	1\$	
			00B3	31 0006A	BRW	16\$	
			68	05 E1 0006D	BBC	#5, (R8), 8\$	
			68	04 E0 00071	BBS	#4, (R8), 11\$	
			3C	01 A8 E9 00075	BLBC	1(R8), 10\$	
			09	09 AE 9F 00079	PUSHAB	ASCII_NAME+1	
				57 DD 0007C	PUSHL	RECADDR	
			0000G CF	02 FB 0007E	CALLS	#2, MAKE_STRING	
		08 AE	AE	50 90 00083	MOVB	R0, ASCII_NAME	
			50	08 AE 9A 00087	MOVZBL	ASCII_NAME, R0	
			50	3B 3A 0008B	LOCC	#59, R0, ASCII_NAME+1	
			02	12 00090	BNEQ	9\$	
			51	D4 00092	CLRL	R1	
			50	09 AE 9E 00094	MOVAB	ASCII_NAME+1, R0	
			51	50 83 00098	SUBB3	R0, RT, ASCII_NAME	
			55	13 A6 9E 0009D	MOVAB	19(R6), R5	
			53	09 AE 9E 000A1	MOVAB	ASCII_NAME+1, R3	
			54	12 A6 9A 000A5	MOVZBL	18(R6), R4	
			52	08 AE 9A 000A9	MOVZBL	ASCII_NAME, R2	
			0000G	30 000AD	BSBW	FMGSMATCH_NAME	
			57	50 E9 000B0	BLBC	R0, 14\$	
				0F 11 000B3	BRB	11\$	
			06	29 000B5	CMPC3	#6, 6(R6), 6(RECADDR)	
			06	4D 12 000BB	BNEQ	14\$	
			0C	A6 B1 000BD	CMPW	12(R6), 12(RECADDR)	
			46	12 000C2	BNEQ	14\$	
			58	68 03 F0 000C4	BBS	#3, (R8), 16\$	
			50	0E A7 32 000C8	CVTWL	14(RECADDR), R0	
			50	04 AE B1 000CC	CMPW	HIGH_VERSION, R0	
			04	10 18 000D0	BGEQ	12\$	
			08	50 B0 000D2	MOVW	R0, HIGH_VERSION	
			08	AC E8 000D6	BLBS	ENTER_MODE, 12\$	
			0E	A6 B5 000DA	TSTW	14(R6)	

			03	12 000DD	BNEQ	12\$: 0555
			5A	59 00 000DF	MOVL	REC_NUM, BEST_REC		: 0558
			50	6E B1 000E2	12\$:	CMPW	LOW-VERSION, R0	
			6E	12 15 000E5	BLEQ	13\$		
			08	50 B0 000E7	MOVW	R0, LOW VERSION		: 0561
			8000	0B AC E8 000EA	BLBS	ENTER MODE, 13\$: 0562
			8F	0E A6 B1 000EE	CMPW	14(R6), #32768		
				03 12 000F4	BNEQ	13\$		
				59 00 000F6	MOVL	REC_NUM, BEST_REC		: 0564
				50 A6 B1 000F9	13\$:	CMPW	14(R6), R0	: 0567
			03	08 AC E9 000FF	BLBC	ENTER MODE, 16\$: 0570
			1D	09 E0 00103	BBS	#9, (R8), 14\$: 0571
			68	5A 00 00107	MOVL	REC_NUM, BEST_REC		
			5A	FF0A 31 0010A	BRW	1\$: 0475
				5A D5 0010D	TSTL	BEST_REC		: 0579
				0F 13 0010F	BEQL	16\$		
				7E D4 00111	CLRL	-(SP)		: 0582
				5A DD 00113	PUSHL	BEST_REC		
			0000G	02 FB 00115	CALLS	#2, DIRGET		
			CF	57 50 00 0011A	MOVL	R0, RECADDR		
				59 5A 00 0011D	MOVL	BEST_REC, REC_NUM		: 0583
			0000G	CF 5B 00 00120	16\$:	FREE_RECORD, FIRST_FREE		: 0590
			0000G	CF AE 32 00125	CVTWL	HIGH_VERSION, HIGHEST_VERSION		: 0591
			0000G	CF 6E 32 0012B	CVTWL	LOW_VERSION, LOWEST_VERSION		: 0592
			0000G	CF 59 00 00130	MOVL	REC_NUM, DIR_RECORD		: 0593
				50 57 00 00135	MOVL	RECADDR, R0		: 0594
				04 00138	RET			: 0595

: Routine Size: 313 bytes. Routine Base: \$CODES + 0000

```
: 283 0596 1
: 284 0597 1 END
: 285 0598 0 ELUDOM
```

PSECT SUMMARY

Name	Bytes	Attributes
\$CODES	313	NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)

Library Statistics

File	----- Symbols -----	Pages	Processing
	Total Loaded Percent	Mapped	Time
\$_\$255\$DUA28:[SYSLIB]LIB.L32;1	18619 18 0	1000	00:02.0

DIRSCN
V04-000

F 5
16-Sep-1984 00:59:21
14-Sep-1984 12:29:30 VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[F11A.SRC]DIRSCN.B32;1 Page 9
(2)

ENT

: COMMAND QUALIFIERS

: BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LISS:DIRSCN/OBJ=OBJ\$:DIRSCN MSRC\$:DIRSCN/UPDATE=(ENH\$:DIRSCN)

: Size: 313 code + 0 data bytes
: Run Time: 00:11.8
: Elapsed Time: 00:29.4
: Lines/CPU Min: 3032
: Lexemes/CPU-Min: 13267
: Memory Used: 160 pages
: Compilation Complete

0165 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

